

[Handwritten signature]
determining whether information corresponding to processor registers for each thread indicates that the thread is running by comparing the information to stored information from a previous interrupt; and

recording time-profiling information for each running thread.

REMARKS

In the Office Action dated February 27, 2002, the Examiner rejected claims 1-25 under 35 U.S.C. section 103(a) as being unpatentable over Doing et al. (U.S. Patent No. 6,018,759) in view of Rosenburg ("How Debuggers Work: Algorithms, Data Structures, and Architecture.") Although Applicant disagrees with the Examiner's rejection, Applicant has amended claims 1, 5, 8, 17 and 24 to further prosecution and to highlight various aspects related to the present invention. Applicant reserves the right to pursue claims without the amendments noted herein.

Regarding claims 1 and 17, these claims recite, among other things, "periodically interrupting execution of all of the threads" and "determining whether register data corresponding to a selected thread has changed from a previous interrupt of all of the threads." Doing et al. and Rosenburg, alone or in combination, fail to teach or suggest a combination of steps including the recited step. The state information maintained by the system described by Doing et al. does not reflect a modification of a register's state from a previous interrupt of all of the threads executing in the system. Instead, the state information maintained by Doing et al. is related to selected events, such as a cache miss, and does not reflect a change from a previous interrupt of all of the threads of a program. Further, Doing et al. and Rosenburg fail to teach or suggest, among other things, periodically interrupting execution of the threads. Although the breakpoint mechanisms discussed by Rosenburg enable a debugging process to interrupt the

FINNEGAN
HENDERSON
FARABOW
GARRETT &
DUNNER LLP

1300 I Street, NW
Washington, DC 20005
202.408.4000
Fax 202.408.4400
www.finnegan.com

execution of a program to service a breakpoint instruction, the execution of the interrupt is dependent on the type of breakpoint instruction included in the program. For example, user breakpoints are defined by a user and are inserted into the code of a program to designate where the program should be interrupted during execution. Thus, only when a program encounters the user-defined breakpoint during execution will the program be interrupted. Further, the various other types of breakpoints discussed by Rosenburg fail to teach periodically interrupting all of the threads of a multi-thread program.

Accordingly, because Doing et al. and Rosenburg, alone or in combination, fail to teach or suggest all of the recitations of claims 1 and 17, Applicant requests that the rejection of the claims under 35 U.S.C. § 103(a) be withdrawn and the claims allowed.

Regarding claims 9 and 25, Applicant submits that the rejection of these claims under 35 U.S.C. § 103(a) is improper because the Examiner failed to address every recitation of these claims. For example, claim 9 recites, among other things, “a processor configured to execute the multi-threaded program, and to periodically interrupt execution of all of the threads” Further, claim 25 recites, among other things, “means for periodically interrupting execution of all of the threads.” The Examiner did not address the periodic interruption of the threads in the previous Office Action. Instead, the Examiner combined the rejection of claims 9 and 25 with the rejection of claims 1 and 17, thus relying on the position that Rosenburg teaches interrupting execution of the threads via the disclosure of breakpoint mechanisms. As previously argued, Doing et al. and Rosenburg, alone or in combination, fail to teach or suggest periodically interrupting execution of all of the threads. Accordingly, Applicant incorporates by reference the above arguments associated with claims 1 and 17 in support of Applicant’s position. Because the Examiner did not address “periodically” interrupting execution of all of the threads in the

FINNEGAN
HENDERSON
FARABOW
GARRETT &
DUNNER LLP

1300 I Street, NW
Washington, DC 20005
202.408.4000
Fax 202.408.4400
www.finnegan.com

rejection of claims 9 and 25, Applicants request that the rejection of these claims be withdrawn and the claims allowed.

Dependent claims 2-4, 10-12, and 18-20 are deemed allowable for at least the same reasons set forth above for claims 1, 9, 17, and 25. Accordingly, Applicant respectfully requests that the rejection of these claims be withdrawn and the claims allowed.

Regarding claims 5, and 21, Applicant has amended this claim to include the recitations of dependent claims 7 and 23, respectively, each of which have been canceled. In the Office Action, the Examiner argued that Doing et al. teaches providing an indication corresponding to a portion of the program containing the selected thread (see Office Action, page 4, lines 15-17). Applicant respectfully disagrees. The bits used by the forward progress state logic described by Doing et al. merely indicate a state of a given thread. They do not, in any interpretation, provide an indication of a portion of a the program the thread was operating. Further, Rosenburg fails to teach, among other things, providing an indication corresponding to a portion of the program containing a selected thread. Accordingly, because neither of the two references cited by the Examiner teach or suggest every recitation of claims 5 and 21, Applicant respectfully requests that the rejection of these claims be withdrawn and the claims allowed.

Dependent claims 6 and 22 are deemed allowable for at least the same reasons set forth above for claims 5 and 21, respectively. Accordingly, Applicant respectfully requests that the rejection of these claims be withdrawn and the claims allowed.

Regarding claims 8 and 24, Applicant incorporates by reference the above arguments associated with claim 1. Accordingly, for at least the same reasons set forth previously for claim 1, Applicants respectfully request that the rejection of claims 8 and 24 be withdrawn and the claims allowed.

Regarding claims 13-16, Applicant incorporates by reference the above arguments associated with claims 9 and 25. Therefore, because the Examiner failed to address the recitation "periodically suspend execution of the multi-threaded program" and that Rosenburg and Doing et al., alone or in combination, fail to teach or suggest the above described recitation, Applicant respectfully requests that the rejection of claims 13-15 be withdrawn and the claims allowed.

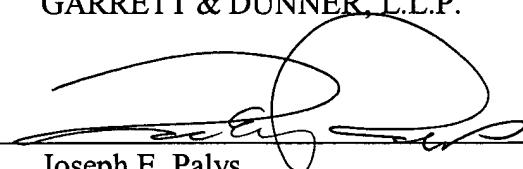
Based on the foregoing arguments, Applicant respectfully requests that the rejection of claims 1-25 under 35 U.S.C. § 103(a) be withdrawn and the claims allowed.

Please grant any extensions of time required to enter this response and charge any additional required fees to our deposit account 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: May 28, 2002

By: 

Joseph E. Palys
Reg. No. 46,508

FINNEGAN
HENDERSON
FARABOW
GARRETT &
DUNNER LLP

1300 I Street, NW
Washington, DC 20005
202.408.4000
Fax 202.408.4400
www.finnegan.com

APPENDIX OF AMENDMENTS TO THE CLAIMS

1. (Twice Amended) A method for time profiling multiple threads of execution corresponding to a program, comprising:
 - periodically interrupting execution of all of the threads;
 - determining whether register data corresponding to a selected thread has changed from a previous interrupt of all of the threads; and
 - providing an indication of the change for the selected thread.

5. (Amended) A method for determining whether a selected thread of execution of a multi-threaded program is running, comprising:
 - suspending execution of the multi-threaded program;
 - retrieving register data corresponding to the selected thread;
 - computing a value based on the register data;
 - comparing the computed value with register information stored following a previous suspension of the multi-threaded program; [and]
 - regarding the selected thread as running if the computed value is different from the previously stored register information; and
 - providing an indication corresponding to a portion of the program containing the selected thread.

8. (Twice Amended) A method for time profiling multiple threads of execution corresponding to a program, comprising:
 - periodically suspending execution of the program;

determining whether information corresponding to processor registers for each thread indicates that the thread is running by comparing the information to stored information from a previous interrupt of all of the threads; and
recording time-profiling information for each running thread.

17. (Twice Amended) A computer-readable medium containing instructions for time profiling multiple threads of execution corresponding to a program, by:
periodically interrupting execution of all of the threads;
determining whether register data corresponding to a selected thread has changed from a previous interrupt of all of the threads; and
providing an indication of the change for the selected thread.

21. (Amended) A computer-readable-medium containing instructions for determining whether a selected thread of execution of execution of a multi-threaded program is running, by:
suspending execution of the multi-threaded program;
retrieving register data corresponding to the selected thread;
computing a value based on the register data;
comparing the computed value with register information stored following a previous suspension of the multi-threaded program; [and]

FINNEGAN
HENDERSON
FARABOW
GARRETT &
DUNNER LLP

1300 I Street, NW
Washington, DC 20005
202.408.4000
Fax 202.408.4400
www.finnegan.com

regarding the selected thread as running if the computed value is different from the previously stored register information; and
providing an indication corresponding to a portion of the program containing the selected thread.

24. (Twice Amended) A computer-readable medium containing instructions for time profiling multiple threads of execution corresponding to a program, by:
periodically suspending execution of the program;
determining whether information corresponding to processor registers for each thread indicates that the thread is running by comparing the information to stored information from a previous interrupt; and
recording time-profiling information for each running thread.

FINNEGAN
HENDERSON
FARABOW
GARRETT &
DUNNER LLP

1300 I Street, NW
Washington, DC 20005
202.408.4000
Fax 202.408.4400
www.finnegan.com